

Demodulation of FM signal using Apache Spark

Pavel Kazenin, www.linkedin.com/in/pavelkazenin/ , pavel.kazenin@gmail.com

Abstract

In this article, a scalable algorithm to demodulate FM signal using Apache Spark framework is described and tested. Implementation of the algorithm and test programs can be found on [GitHub](#).

General equations and formulas

In FM-modulated signal, message $s(t)$ is coded by modulating carrier frequency $x(t) = A \cdot \sin(\omega_T t + S(t))$, $S(t) = k_{mod} \int s(t) dt$. Typically change rate of $S(t)$ is 1000 times less than carrier frequency ω_T and during sampling period the difference $\Delta \omega_H$ between message frequency ω_H and carrier frequency ω_T is approximately constant so $S(t) \approx \Delta \omega_H t + \phi$ where ϕ is a constant phase shift angle:

$$x(t) = A \cdot \sin(\omega_H t + \phi)$$

Let $y(t) = A \cdot \sin(\omega_R t)$ be a reference signal of frequency ω_R . For every measured signal value of $x(t_i)$ a reference signal value of $y(t_i)$ can be obtained. We now have parametric equations with time t being a parameter:

$$\begin{cases} x(t) = A \cdot \sin(\omega_R t + \Delta \omega t + \phi) \\ y(t) = A \cdot \sin(\omega_R t) \end{cases}$$

Phase shift angle between message and reference signals $\Delta \omega t + \phi$ can be expressed in terms of $x(t)$ and $y(t)$:

$$\theta(t, \phi) = |\sin(\Delta \omega t + \phi)| = |x(t) \sqrt{1 - y^2(t)} \pm y(t) \sqrt{1 - x^2(t)}| \quad (1)$$

For $\omega_H = \omega_R$ frequency difference $\Delta \omega = 0$ and parametric equations $x(t), y(t)$ form a Lissajous ellipse. The sign in the expression (1) corresponds to one of two Lissajous ellipses on (x, y) plane where $x(t), y(t)$ resides. For N samples and, thus, $2N$ values of θ (N values of θ^+ for “+” sign and N values of θ^- for “-” sign), at least N values reside on the same ellipse and have the same values of θ_0 since for coherent signals phase shift is time invariant: $\theta_0(t, \phi) = \text{const}(t)$, $\Delta \theta_0 = 0$:

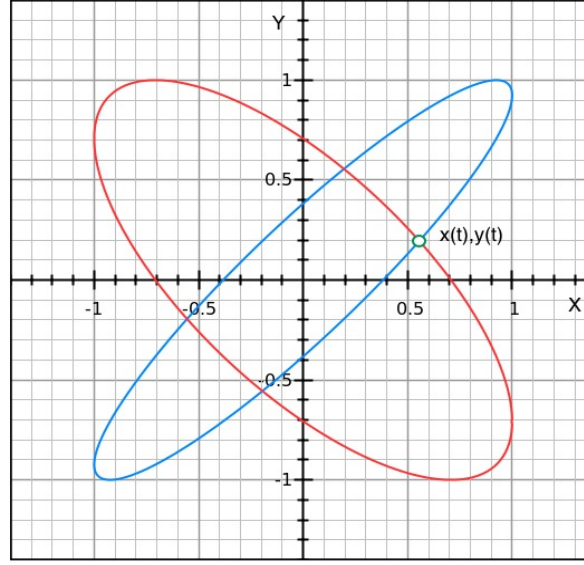


Figure 1. Two Lissajous ellipses crossing the same point on (X,Y) plane, one ellipse corresponds to “+” sign and the other one to “-” sign in formula (1).

When $\omega_H \neq \omega_R$, $\Delta \omega = |\omega_H - \omega_R|$ and $\Delta \theta_0 = |\theta_0(t_2, \phi) - \theta_0(t_1, \phi)| = |\sin(\Delta \omega t_2 + \phi) - \sin(\Delta \omega t_1 + \phi)|$,

$$\Delta \theta_0 = |2 \cdot \cos\left(\frac{\Delta \omega}{2}(t_2 + t_1) + \phi\right) \cdot \sin\left(\frac{\Delta \omega}{2}(t_2 - t_1)\right)|$$

We will state that $\omega_H \approx \omega_R$ with acceptable accuracy $\Delta \omega$ if for N samples at least for N values of θ_0 , the inequation

$$\Delta \theta_0 \leq \frac{1}{B} \quad (2)$$

holds for some integer value B . If sampling measurements are randomly taken within period of ΔT , the average distance between any two measurements $\Delta t = \Delta T/2$. For small arguments $\sin x \approx x$. First multiplicand, the cosine function of random argument, diverges, however the average absolute value of cosine is $2/\pi$. The acceptance criteria then reduces to $\Delta \omega \Delta T \leq \pi/B$, or in terms of cycles per second,

$$\Delta f \Delta T \leq \frac{1}{2B} \quad (3)$$

That is, the higher accuracy and the smaller sampling period, the larger B has to be to satisfy criteria (3).

The number of samples N is chosen large enough so the probability that at least one cosine value is greater than average $2/\pi$ is big enough. For $N=10$, $p=1-0.5^N=0.99902$.

Say, carrier frequency is 100 kHz, bandwidth is 100 Hz, sampling period is 100 ms corresponds to 10 cycles at bandwidth frequency, the accuracy is 0.01%, $\Delta f = 0.01$ Hz, $\Delta T = 0.1$ s $\rightarrow B > 500$

It is easy to notice that B is a number of histogram bins on the interval $[0, 1]$, each bin has a width of $1/B$, and the criteria (2) means that at least N values of θ_0 fall into the same bin. Let ω_{R1} and ω_{R2} be two neighbor reference frequencies and $\omega_H = \frac{\omega_{R1} + \omega_{R2}}{2}$. If we choose B too large, the criteria (2) does not hold for neither ω_{R1} nor ω_{R2} , that is, even if message signal frequency lies between two reference frequencies, it can't be identified by calculating and comparing $\Delta \theta_0$. The difference between ω_{R1} and ω_{R2} can't exceed the accuracy $\Delta \omega$ and B is limited from above by the same value $\frac{1}{2\Delta f \Delta T}$. Thus, the optimal value of B so the criteria (2) still holds and maximum accuracy is achieved is

$$B = \frac{1}{2\Delta f \Delta T} \quad (4)$$

Algorithm description

The idea behind the algorithm is to guess frequency of the message signal $x(t) = A \cdot \sin(\omega_H t + \phi)$ by calculating $2N$ values of θ for reference signal $y_i(t)$ of frequency ω_{Ri} $y_i(t) = A \cdot \sin(\omega_{Ri} t)$ and checking if at least N values of θ fall into the same bin $\left[\frac{j}{B}, \frac{j+1}{B}\right]$ where j is a bin number $0 \leq j < B$.

Total number of probing reference signals K depends on desired accuracy $\Delta f = \frac{\Delta F}{K}$ where ΔF is a bandwidth. The number of bins B is computed using formula (4). Because calculation of θ for each reference signal $y_i(t)$ is independent, the algorithm is easily parallelizable and scalable.

Below is a pseudocode of the algorithm:

1. Read N samples of FM-modulated signal from data stream.

output: N tuples (timestamp, message signal)

2. Compute θ^+ and θ^- values for K probing frequencies

output: $2NK$ key-value pairs:

key = reference frequency,

value = tuple (reference frequency, timestamp, θ^+ , θ^-)

3. Aggregate by reference frequency and compute min and max timestamp values, and identify a bin with maximum number of samples maxCounter which θ value falls into.

output: K key-value pairs:

key = reference frequency

value = tuple (maxCounter, number of samples N , min timestamp, max timestamp)

4. Filter out records with maxCounter < number of samples.
5. Aggregate and average all frequencies which passed filter.

output: average timestamp, average message signal frequency

6. Save average timestamp and average message signal frequency in a file or pipe it to other program for further processing, analysis, differentiation, etc.

Experimental results

In the experiments, two signals have been simulated with random phase shifts and timestamps, one for frequency matching reference frequency, and the other for frequency located between two neighbor reference frequencies. The precision of the simulated signals and timestamps has been chosen to match accuracy of 20-bit ADC. In each experiment, 20 trials have been made for $B = \{100, 200, 300, \dots\}$. Average message signal frequency and standard error have been calculated.

Table 1. Average measured signal frequency and standard error.

Accuracy 0.01 Hz, sampling period 10 ms, number of trials 20.

Number of bins	Signal frequency 100,000.070 Hz		Signal frequency 100,000.075 Hz	
B	Measured frequency	Standard error	Measured frequency	Standard error
100	99999.9570	0.6373	99999.7122	2.0528
200	99999.9655	0.3864	99999.9354	0.3583
300	99999.8952	0.3816	99999.9793	0.4436
400	99999.9768	0.2667	100000.1319	0.4825
500	100000.0931	0.1162	100000.0320	0.2022
600	100000.0649	0.0205	100000.0739	0.2095
700	100000.0718	0.0129	100000.0691	0.0172
800	100000.0640	0.1930	100000.0761	0.0275
900	100000.1103	0.2633	100000.0998	0.1126
1000	100000.0776	0.1058	100000.0629	0.0497
1100	100000.0714	0.0082	100000.0324	0.1646
1200	100000.0696	0.0113	100000.0734	0.0052
1300	100000.0651	0.2215	100000.0733	0.0072
1400	100000.1178	0.2156	100000.0747	0.0107
1500	100000.0696	0.0036	100000.0230	0.2224
1600	100000.0695	0.0074	100000.0763	0.0050

1700	100000.0463	0.1093	100000.0731	0.0056
1800	100000.0157	0.1482	100000.1302	0.1533
1900	100000.0611	0.0257	100000.0870	0.0657
2000	100000.0697	0.0056	100000.0706	0.0191
2100	100000.0495	0.0998	100000.0740	0.0097
2200	100000.0734	0.0174	100000.0683	0.1068
2300	100000.0955	0.1135	100000.0760	0.0086
2400	100000.0699	0.0039	100000.0765	0.0065
2500	100000.0695	0.0031	N/A	N/A

The results show that occasionally, approximately once per 20 trials, the algorithm calculates severely incorrect frequency. Below is typical output of the program. Numbers in red highlight “outliers”. The “cut-off” number of bins $B = 2500$ turns out to be higher than estimated by formula (4).

Table 2. Results of 20 trial measurements.

Signal frequency 100000.070 Hz, Accuracy 0.01 Hz, sampling period 10 ms, number of bins 2200.

Trial	Measured frequency	Timestamp, ms
1	100000.0700	5.3600
2	100000.0699	2.9686
3	100000.0650	5.2307
4	100000.0650	4.8241
5	100000.0700	5.1301
6	100000.0700	4.8447
7	100000.0700	5.3027
8	100000.0700	5.1634
9	100000.0700	3.2580
10	100000.0700	4.7373
11	100000.0450	4.7755
12	100000.0650	4.1329
13	100000.0849	4.8820
14	100000.0700	5.6797
15	100000.0700	4.8737
16	100000.0700	4.9840
17	100000.0700	4.8767
18	100000.1336	6.4105
19	100000.0600	4.8777
20	100000.1050	4.6720